

Keijo Pitkäniemi

# Asiakasmäärälaskuri ja siihen liittyvä tietokanta-analyysi



Tradenomi

Tietojenkäsittelyn ko

Kevät 2018



KAJAANIN  
AMMATTIKORKEAKOULU  
UNIVERSITY OF APPLIED SCIENCES

## TIIVISTELMÄ

**Tekijä:** Pitkäniemi Keijo

**Työn nimi:** Kävijälaskuri ja siihen liittyvä tietokanta- analyysi

**Tutkintonimike:** Tradenomi, tietojenkäsittely

**Asiasanat:** tietokanta, asiakaslaskuri

Tässä opinnäytetyössä suunniteltiin ja rakennettiin käytännöllinen tietokantaympäristö baari- ja yökerhotoimijoille, sekä mietittiin järjestelmän mahdollisia hyötyjä sitä käyttäville yrityksille. Alussa on tarkastelua siitä, millä tavoin nykyään kerätään asiakasmääriä kohdeyrityksissä, mutta työn pääpaino on toimivan asiakasmäärien keräämisjärjestelmän suunnittelu ja rakennus.

Ennen järjestelmän rakentamista se suunniteltiin teoreettisella tasolla. Teoreettisen järjestelmän pohjalta suunniteltiin ja rakennettiin käytännön toteutus. Tosin käytännön toteutuksesta tuli huomattavasti erilainen verrattuna teoreettiseen suunnitelmaan.

Järjestelmä rakennettiin selaimella käytettäväksi. Sen rakentamiseen käytettiin cPanel-alustaa ja se toteutettiin yhden verkkoalueen sisälle. Järjestelmään tehtiin kirjautumisen, asiakkaiden lisäämisen ja kerätyn tiedon tarkastelun osiot.

Tietoturvariskejä kuvaillaan opinnäytetyön loppuosassa. Huomioon otettiin riskejä, kuten SQL-injektiot, tiedon salaus ja fyysisen laitteiston tuhoutuminen. Näihin esitetään myös ratkaisuja, joista varmuuskopiointia käsitellään tarkimmin.

## ABSTRACT

**Author:** Pitkääniemi Keijo

**Title of the Publication:** Customer Counter and Related Database Analysis

**Degree Title:** Bachelor of Business, Computing

**Keywords:** customer counting, database

This Bachelor's thesis consists of constructing a useful database environment for companies in bar and nightclub business and researching possible benefits using this environment will bring to those companies. There are some points on how the counting of customers is done currently, but the main focus of this work is to plan and produce a working environment for that.

The idea of this kind of system came from a bar and night club doorman. By discussing this matter with him, some ways to significantly improve the data collection on customer arrivals were invented.

Planning an ideal way to make this kind of environment and how to make companies use it are one part of this work. The other part is actually building a working environment that can be used for the planned purposes. There is some consideration about security issues, too, but most of them are not relevant during this work.

## SISÄLLYS

|  |    |
|--|----|
| 1 JOHDANTO.....                                  | 1  |
| 2 TARVE ASIAKASMÄÄRÄTIETOKANNALLE .....          | 2  |
| 2.1 Tarve .....                                  | 2  |
| 2.2 Tavoitteet .....                             | 3  |
| 3 JÄRJESTELMÄN TEKNINEN TOTEUTUS .....           | 4  |
| 3.1 Teoria .....                                 | 4  |
| 3.1.1 Tiedon keräys.....                         | 4  |
| 3.1.2 Tiedon kulku.....                          | 5  |
| 3.1.3 Tiedon säilytys .....                      | 7  |
| 3.1.4 Tiedon käyttö.....                         | 8  |
| 3.2 Käytäntö .....                               | 9  |
| 3.2.1 Tiedon keräys.....                         | 10 |
| 3.2.2 Tiedon käyttö.....                         | 15 |
| 3.2.3 Tiedon kulku.....                          | 18 |
| 4 TIETOTURVA.....                                | 19 |
| 4.1 Lailliset kysymykset.....                    | 19 |
| 4.2 Palveluntarjoajan ja -saajan näkökulma ..... | 19 |
| 4.3 Mahdollisia riskejä ja niiden torjuntaa..... | 20 |
| 4.4 Tietokannan varmuuskopiointi.....            | 20 |
| LÄHTEET .....                                    | 22 |

## SYMBOLILUETTELO

Bluetooth – langaton tiedonsiirtojärjestelmä

cPanel – verkkosivujen luomiseen suunniteltu alusta

Happy Hour – anniskelupaikoissa tietty ajanjakso, jolloin juomia saa halvemmalla

MySQL - tietokantaohjelmisto

PHP – komentosarjakieli, jota käytetään verkkosivujen toiminnallisuudessa

## 1 JOHDANTO

Ajatus asiakasmääriä keräävän järjestelmän tekemiseen lähti keskustelusta alalla toimivan tuttavän kanssa. Aihetta käsiteltyämme ilmeni, että ainakin monien baarien ja yökerhojen ovilla asiakasmäärät kerätään vanhanaikaisesti. Vanhanaikaisella tarkoitan järjestelmää, jossa tieto ei automaattisesti päädy sähköiseen tietokantaan.

Ideaksi muodostui rakentaa tietokanta ja tiedonkeruujärjestelmä, jolla asiakasmäärien tarkastelu olisi tehokasta ja helppoa. Alueen laajuudeksi rajautui tietojen keräämisen mahdollistavan sovelluksen, tiedon varastoinnin ja graafisen esittämisen tarjoaminen yrityksille. Tätä laajempi järjestelmä ulottuisi alueille, jotka yrityksillä on jo automatisoitu nykyaikaisella tavalla. Pyrkimyksenä on välttää tätä tapahtumasta, koska valmiin käytössä olevan järjestelmän syrjäyttäminen ei ole tämän työn tavoitteena. Lisäksi tästä työstä syntyvän tuotteen saaminen yritysten käyttöön vaikeutuisi entisestään, jos olemassa olevia tietojärjestelmiä pitäisi sopeuttaa meidän tarjoamaamme tuotteeseen.

Suunnitellun tietojärjestelmän laajentaminen tulevaisuudessa jää opinnäytetyössä pohdinnan asteelle. Mahdollinen menestys yritysmarkkinoilla voisi avata ovia laajentamiseen, mutta ennen sitä on saatava vakaa asema tavoitellussa ympäristössä.

Järjestelmä rakentuu kolmesta pääkohdasta: tiedon keräämisestä, säilytyksestä ja graafisesta esittämisestä. Tavoitteena on näiden toimintojen mahdollisimman hyödyllinen ja helppo käytettävyys. Järjestelmän rakennuttua käytettävään muotoonsa se olisi mahdollista antaa yritysten testattavaksi, jonka jälkeen tuotteen voisi viimeistellä ja viedä markkinoille.

## 2 TARVE ASIAKASMÄÄRÄTIETOKANNALLE

### 2.1 Tarve

Yritys laskee asiakasmääriään selvittääkseen, milloin liiketoiminta on sujuvaa ja milloin siinä olisi parantamisen varaa. Mitä tarkempaa tietoa yritys saa, sitä paremmin se pystyy suhteuttamaan käytettävät resurssinsa toimintansa optimointiin. Yritys voi koettaa lisätä kävijämääriään järjestämällä 'happy houreja' ja erilaisia tapahtumia [1]. Asiakasmääriä tarkkailemalla yritys pystyy arvioimaan näiden vaikutusta kävijämääriin.

Nykyisellään monet yökerhot ja baarit kirjaavat saapuvat asiakkaat ylös klikkerillä. Klikkeri on mekaaninen laite, jossa olevaa nappia painamalla siinä oleva lukema nousee yhdellä, (kuva 1).



Kuva 1. Klikkeri

Kerätty tieto siirretään manuaalisesti illan päätteeksi eteenpäin. Näin tehdyssä työssä tulee ylimääräisiä työvaiheita ja osa tiedosta ei jää talteen. Esimerkiksi aika jona asiakas saapuu, ei mitenkään jää talteen klikkeriä käytettäessä.

Suurimmalta muulta osalta yökerhojen ja baarien toiminnan tarkkailu on sähköisesti automatisoitua, joten oletettavissa on, että myös automatisoitu kävijämäärän seuranta olisi mieleen kyseisten liikkeiden yrittäjille. Tilastokeskuksen mukaan

majoitus- ja ravintolatoiminnalla vain 21 % yrityksistä on käytössä asiakkuuden hallinnan tietojärjestelmä [2]. Vaikka tämän opinnäytetyön tarjoama ratkaisu ei varsinaisesti katakaan koko asiakkuuden hallinnan tietojärjestelmän määritettä, kertoo alhainen prosentti samantyyppisessä asiassa kuitenkin siitä, että oikeanlaiselle järjestelmälle olisi tilaa markkinoilla.

## 2.2 Tavoitteet

Opinnäytetyön tuottaman järjestelmän tarkoituksena on antaa kohderyhmään kuuluville yrityksille mahdollisimman käytännöllinen tapa seurata asiakasvirtaansa puhtaasti määrällisestä näkökulmasta tarkasteltuna. Kerättävän tiedon laatu on pidetty suppeana syistä, joita käsitellään myöhemmissä kappaleissa tarkemmin. Yleistettynä kuitenkin voisi tiivistää syyt liittymään järjestelmän rakentamisen, ylläpidon ja käytettävyyden helppouteen.

Helppokäyttöisyyden tavoittelemisen on jo itsessään hyvä asia järjestelmää suunniteltaessa. Tässä tapauksessa sen päätarkoitus on kuitenkin helpottaa järjestelmän saamista yritysten käyttöön. Mikäli järjestelmän käyttö vaatisi henkilöstön kouluttamista tai laajoja hankintoja, yritysten saaminen tarpeeksi kiinnostuneeksi tuotteesta, jotta se päätyisi käyttöön asti, vaikeutuisi huomattavasti.



### 3 JÄRJESTELMÄN TEKNINEN TOTEUTUS

Tässä kappaleessa käydään läpi asiakasmäärätietokannan toiminta ja rakentaminen sekä teorian että käytännön tasolla. Teoriaosuus selittää periaatteita ja tapoja, joilla tieto kerätään ja käytetään. Käytännön osuudessa esitellään järjestelmän rakentaminen ja käyttö vaiheittain.

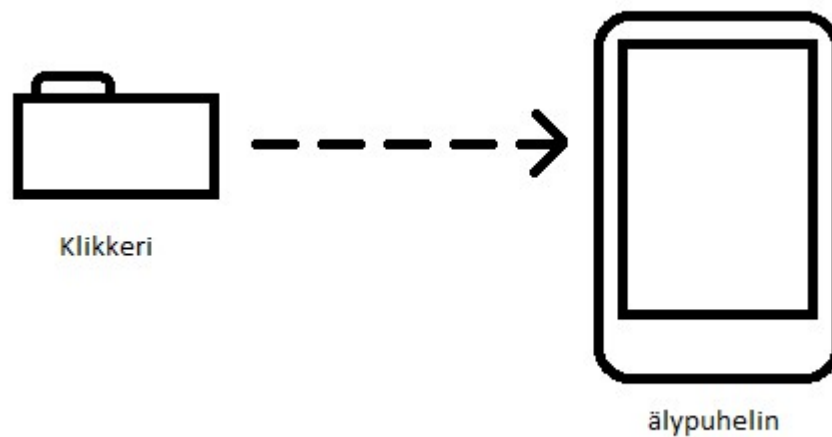
#### 3.1 Teoria

##### 3.1.1 Tiedon keräys

Nykyään suuri osa yökerhoista ja baareista kerää tiedot asiakasmäärästään klikkerillä. Illan lopuksi tiedot siirretään manuaalisesti eteenpäin. Nykyinen käytäntö on kuitenkin hyvin yksinkertainen ja helppokäyttöinen, joten uuden keräysmenetelmän tavoitteena on tuntua itse keräämisen osalta samalta kuin vanhallaakin menetelmällä.

Järjestelmän tiedonkeruuta varten kehitetään mobiilisovellus. Sovellus ladataan normaaliin tapaan älypuhelimeen. Sovelluksen käyttämiseksi täytyy luoda käyttäjätunnus järjestelmään. Kun sovelluksen käyttö aloitetaan, siihen kirjaututaan sisään työpaikan tunnuksilla. Näin kerätty tieto saadaan tallennettua tietokantaan oikean yrityksen tietoihin. Kun sovellus on käynnissä ja siihen on kirjaututtu, on tiedon kerääminen helppoa. Painetaan vain yhtä nappia sovelluksessa ja tieto saapuneesta asiakkaasta ja kellonajasta, jolloin asiakas on saapunut, tallennetaan automaattisesti.

Mahdollisesti voitaisiin rakentaa myös klikkeri, joka yhdistetään Bluetoothilla älypuhelimeen. Tällä tavoin tiedonkeruujärjestelmä käytön aikana tuntuisi identtiseltä sitä käyttävälle henkilölle. Käytännössä tässä opinnäytetyössä ei kuitenkaan tulla rakentamaan minkäänlaista uutta laitetta, joten tämä idea on esitetty enemmänkin tulevaisuutta ajatellen. Bluetooth-klikkerin ja älypuhelimien yhteys on esitetty kuvassa 2.

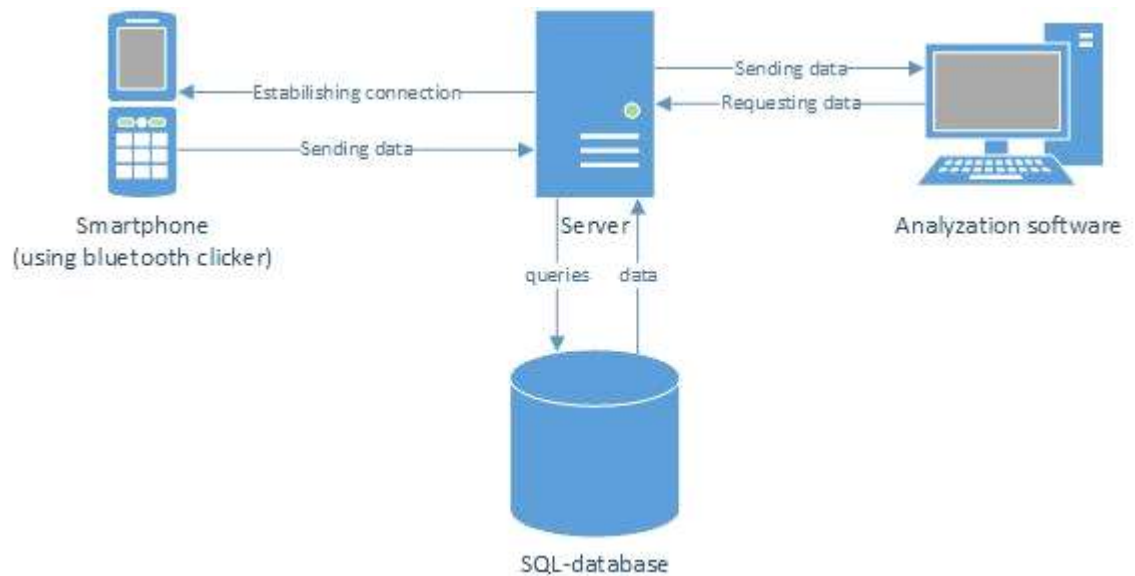


Kuva 2. Bluetooth-klikkeri

### 3.1.2 Tiedon kulku

Tietoa siirrettäessä on otettava huomioon lukuisia tekijöitä. Järjestys jossa tieto siirtyy ja paikka johon se siirretään, riippuu hyvin paljon tietoturvaan liittyvistä tekijöistä. Jokainen välivaihe tiedonsiirrossa on kohta, jossa jokin voi mennä vikaan. Muita kuin tekniisiin rajoitteisiin liittyviä välivaiheita tulisi tiedon kulussa siis rajoittaa mahdollisuuksien mukaan.

Tiedonkulkuun kuuluu neljä eri sijaintia. Sovellus joka kerää tiedon on ensimmäinen osa. Sovellus lähettää tiedon palvelimelle, joka on toinen osa. Tieto tallennetaan tietokantapalvelimelle, järjestelmän kolmanteen osaan. Kun kerättyä tietoa halutaan tarkastella, lähettää tiedon tarkasteluun tehty sovellus, joka on järjestelmän neljäs sijainti, pyynnön palvelimelle. Palvelin tekee kyselyn tietokannalle, joka lähettää tiedon palvelimen kautta takaisin sovellukselle. Näiden sijaintien välillä tiedon on kuljettava turvallisesti ja tehokkaasti, jotta järjestelmä olisi käytännöllinen. Tiedonkulku on esitetty kuvassa 3.



Kuva 3. Tiedon kulku

Tiedonkulun ensimmäinen osa on sen keräys. Sovellus kerää tiedon asiakkaan saapumisesta napin painalluksella ja merkitsee tietoon mukaan tarkan kellonajan, muodostaen niistä datapaketin. Käytännössä pakettiin jäävä tieto on vain kellon-aika, koska tieto siitä, että joku on saapunut paikalle, sisältyy jo itsessään paketin muodostamiseen ja lähettämiseen. Sovelluksesta tieto siirtyy palvelimelle. Jotta tieto saadaan siirrettyä turvallisesti ja oikeaan paikkaan, on sovelluksella kirjauttava käytön aluksi järjestelmään käyttäjätunnuksilla. Näin ollen palvelin saa myös tiedon, siitä kenelle kerätty tieto kuuluu ja osaa tallentaa sen oikeaan paikkaan tietokannassa.

Saatuun sovelluksen lähettämän datapaketin palvelin tarkistaa, että saatu datapaketti ei sisällä tietokannalle vahingollista dataa. Tietokantaa vahingoittavan tiedon lisäämistä tietokantaan lähetettävään pakettiin kutsutaan SQL-injektioksi [3]. Toinen asia, jonka palvelin tarkistaa jo aiemmin, on käyttäjän tunnistaminen. Kerran tämän tehtyään palvelin pitää muodostetun istunnon voimassa, kunnes sovellus kirjautuu ulos järjestelmästä. Näitä käsitellään tarkemmin käytännön toteutusta ja tietoturvaa esittelevissä osuuksissa.

Kun palvelin on tarkastanut, että datapaketti on kunnossa, siirtyy se seuraavaan vaiheeseen. Palvelin muokkaa vastaanotetusta paketista tietokantapalvelimelle

esitettävän kyselyn. Kyselyyn tulee alkuperäisen tiedon lisäksi tieto siitä, kuka datapaketin alun perin lähetti. Kun kysely on tehty tietokantapalvelimelle, on tieto vihdoin tallentunut järjestelmään.

Tietokantaan tallennettu tieto on tehty käytettäväksi, joten on tärkeää myös suunnitella sen saaminen käyttöön turvallisesti ja tehokkaasti. Tietojen tarkasteluun on suunniteltu toinen sovellus, joka näyttää kerätyn tiedon graafisesti. Näin haluttu tieto on helppo saada havainnollistettua sitä tutkivalle henkilölle.

Tiedon graafiseen esittämiseen suunniteltu sovellus vaatii myös kirjautumisen käyttäjätunnuksilla. Sovellus pyytää palvelimelta tiedot, jotka se aikoo esittää graafisessa muodossa. Palvelin tietää kirjautumistiedoilla, mihin tietokannan osuuteen se antaa käyttäjän tehdä kyselyitä. Sovelluksen käyttäjä ei myöskään tässä tapauksessa pääse tekemään suoraan kyselyitä tietokantaan. Haluttu tieto on esitetty palvelimelle, joka sitten tekee tietokantaan kyselyn annettujen tietojen perusteella.

Kerätystä yksinkertaisesta tiedosta saadaan tämän ketjun kautta helposti ja monipuolisesti hyödynnettävää tietoa. Tieto myös pysyy turvassa ja päättyy vain henkilöille, joille se kuuluu.

### 3.1.3 Tiedon säilytys

Turhan tiedon säilyttäminen ei hyödytä asiakasta mitenkään ja vie turhaa tilaa palveluntarjoajan järjestelmissä. Suunnittelemalla alusta asti huolellisesti, kuinka pitkään ja missä muodossa tieto säilytetään, saadaan järjestelmä, jossa tieto on asiakkaan ja palveluntarjoajan kannalta mielekästä. Tiedon tarpeellisuutta tulevaisuudessa ei kuitenkaan voida täydellä varmuudella usein tietää. Tästä syystä on tehtävä realistinen ja aiempaan tietoon perustuva arvio tiedon säilyttämisen tärkeydestä.

Lisäksi tiedon säilyttämisessä on huomioitava, että tieto varmasti säilyy. Varmuuskopioinnin suunnittelu ja mitoitus järjestelmään on tehtävä huolella. Kerätty tieto vie kuitenkin suhteellisen vähän tilaa, joten tilanpuutteen ei pitäisi tulla esteeksi

varmuuskopiointia suunnitellessa. Tiedon säilöntään liittyy tiettyjä riskejä, joita tulee huomioida, mutta niiden tarkempi käsittely on muiden tietoturva-asioiden yhteydessä.

Periaatteessa tietoa voisi varastoida todennäköisesti monellekin asiakkaalle vuosien ajan, mutta käytännöllisintä olisi rajoittaa tiedon varastointi asiakkaan tarpeiden mukaan. Yksinkertaisin malli, jolla rajoittaa tiedon varastointia, on veloittaa asiakkaalta enemmän pidemmästä säilytysajasta.

### 3.1.4 Tiedon käyttö

Asiakkaalle järjestelmän kiinnostavin osa on tietenkin se, miten kerättyä tietoa voidaan hyödyntää. Asiakasmääriä tarkkailemalla voi havaita, missä ravintola-alan yrityksellä menee hyvin ja mitä kehitettävää mahdollisesti on. Sovellus, jolla kerätty tieto sitten esitetään sitä tutkivalle henkilölle, voi olla hyvinkin yksinkertainen.

Asiakasmäärien esittäminen graafisilla kuvaajilla ei tietenkään ole mitään uutta näille yrityksille, mutta kuuluu oleellisena osana niille tarjottavaan järjestelmään. Joillakin yrityksillä saattaa olla jo käytössä ohjelmia, joilla tarkkaillaan asiakasmäärien kehitystä. Mikäli sellainen järjestelmä jo löytyy yritykseltä, on tässä opinnäytetyössä tehtävää järjestelmää muokattava näille sopivammaksi.

Kerättyä tietoa tutkimalla yrityksen toimintaan liittyviä päätöksiä tekevä henkilö pystyy arvioimaan henkilöstön käyttöä tiettyihin aikoihin hyvinkin tarkasti. Lisäksi voidaan tarkkailla, miten erilaiset järjestetyt tapahtumat vaikuttivat asiakasvirtaan. Esimerkiksi jos jokin tietty tapahtuma tuo paljon asiakkaita, sen uusiminen on yritykselle kannattavaa.

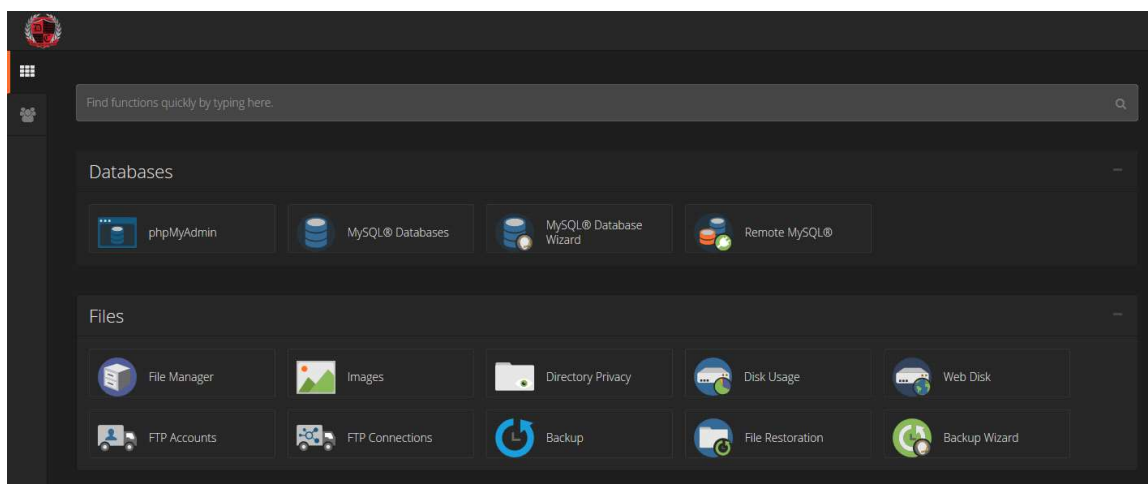
Näillä tavoin järjestelmäämme käyttävä yritys pystyy tekemään säästöjä ja voittoa. Realistinen arvio siitä, kuinka paljon rahallista hyötyä yritys saa käyttämällä suunniteltua järjestelmää selviää vasta, kun se on saatu testikäyttöön johonkin kohteeseen.

### 3.2 Käytäntö

Järjestelmän toteutustapa muuttui huomattavasti työn edetessä. Sen alkuperäinen sovellukseen pohjaava käytötapa vaihtui selainpohjaiseen ratkaisuun. Käytännössä muutos ei kuitenkaan vaikuttanut järjestelmän käyttötapaan lainkaan.

Aiemmin sovelluksissa tapahtuvat asiat tehdään nyt yhden verkkoalueen alaisuudessa. Tämä helpottaa tiedonsiirtoa ja pitää järjestelmän helposti hallittavana. Alustana käytin cPanel-infrastruktuuria.

cPanel on webhotellityyppinen alusta, jolla voidaan tehdä ja hallita verkkosivuja sekä niihin liittyviä järjestelmiä kuten tietokantoja. cPanel-alustan käytön opettelu vei jonkin verran aikaa, mutta nopeutti itse järjestelmän rakentamista huomattavasti. cPanelin etusivu on näytetty kuvassa 4.

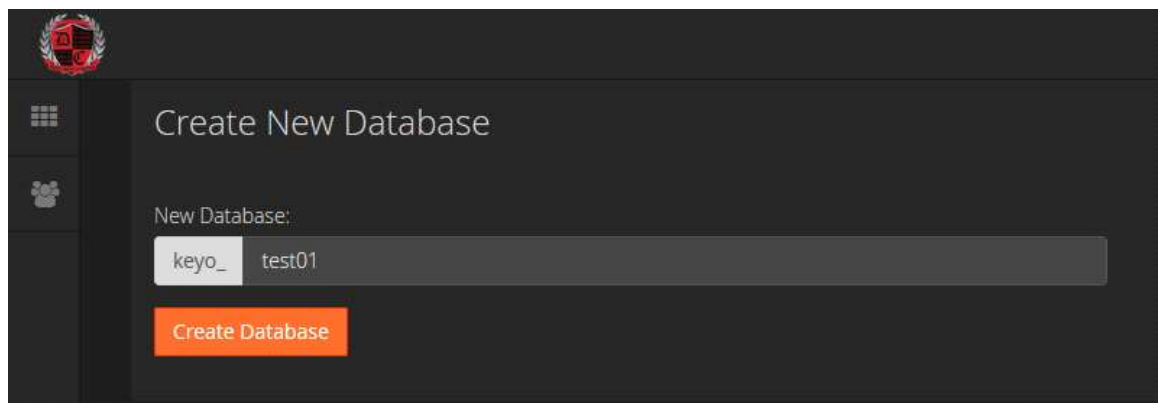


Kuva 4. cPanel, etusivu

### 3.2.1 Tiedon keräys

#### 3.2.1.1. Tietokannan luonti

Järjestelmän rakentamisen ensimmäinen vaihe on tehdä tietokanta, johon tieto tallentuu. cPanelista löytyy MySQL-tietokannan rakennustyökalu. Tietokanta luodaan yksinkertaisesti antamalla tietokannalle nimi ja valitsemalla "luo tietokanta". Tässä tapauksessa nimesin tietokannan test01-nimiseksi, (kuva 5).



Kuva 5. Tietokannan luonti

Sen jälkeen tietokantaan luodaan käyttäjä, jolle asetetaan oikeudet sen mukaan, mitä käyttäjä tulee tietokannassa tekemään. Tässä tapauksessa käyttäjän nimeksi annoin "testuser01". Käyttäjä luodaan antamalla käyttäjälle nimi, salasana ja valitsemalla "luo käyttäjä". Salasana olisi myös mahdollista tehdä käyttämällä automaattista salasanan luontia. Käyttäjän luonti on esitetty kuvassa 6 ja oikeuksien asettaminen on esitetty kuvassa 7.

MySQL Users

Add New User

Username  
keyo\_ testuser01

Password  
.....

Password (Again)  
.....

Strength ⓘ  
Strong (17/100)

Create User

Password Generator

Kuva 6. Tietokantakäyttäjän luonti

Manage User Privileges

User: **keyo\_testuser01**

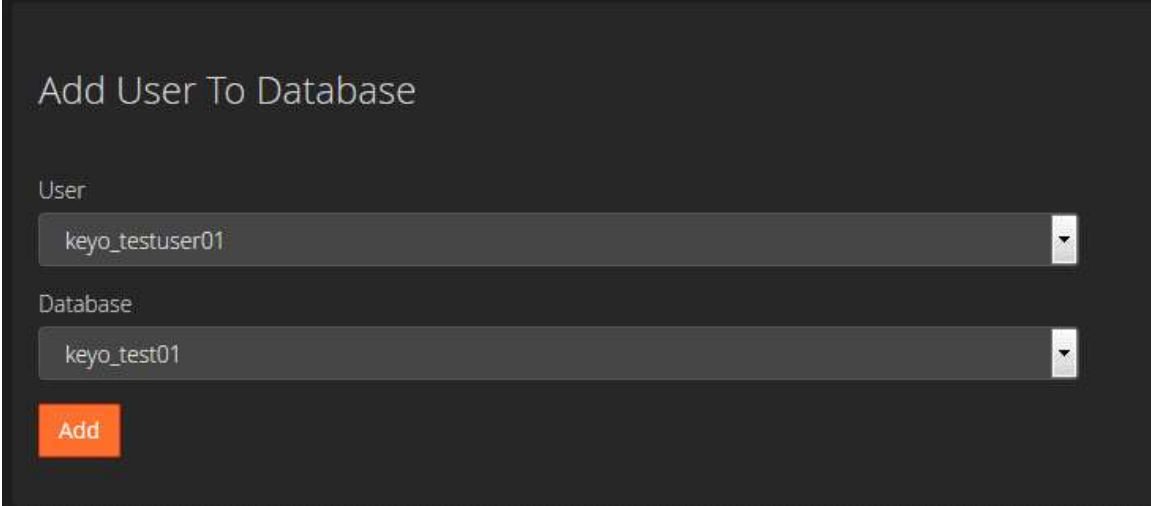
Database: **keyo\_test01**

|  |  |
|--|--|
| <input type="checkbox"/> ALL PRIVILEGES          |  |
| <input type="checkbox"/> ALTER                   | <input type="checkbox"/> ALTER ROUTINE     |
| <input type="checkbox"/> CREATE                  | <input type="checkbox"/> CREATE ROUTINE    |
| <input type="checkbox"/> CREATE TEMPORARY TABLES | <input type="checkbox"/> CREATE VIEW       |
| <input type="checkbox"/> DELETE                  | <input type="checkbox"/> DROP              |
| <input type="checkbox"/> EVENT                   | <input type="checkbox"/> EXECUTE           |
| <input type="checkbox"/> INDEX                   | <input checked="" type="checkbox"/> INSERT |
| <input type="checkbox"/> LOCK TABLES             | <input type="checkbox"/> REFERENCES        |
| <input checked="" type="checkbox"/> SELECT       | <input type="checkbox"/> SHOW VIEW         |
| <input type="checkbox"/> TRIGGER                 | <input type="checkbox"/> UPDATE            |

Kuva 7. Tietokantakäyttäjän oikeuksien asettaminen

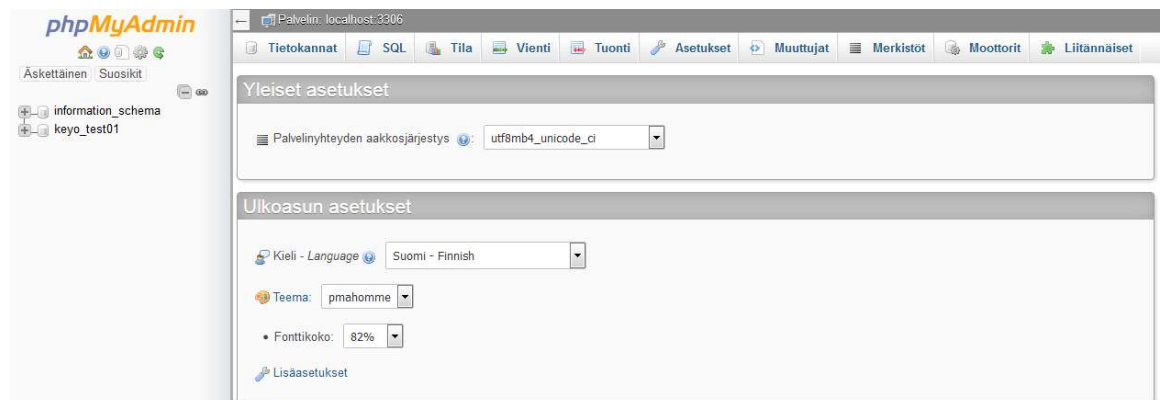


Tehty käyttäjä lisätään tietokantaan valitsemalla tietokanta, johon käyttäjä lisätään. Tämä tapahtuu kohdassa "lisää käyttäjä tietokantaan" ja valitsemalla "lisää", (kuva 8).



Kuva 8. Käyttäjän lisääminen tietokantaan

Seuraava vaihe on tehdä tietokannan rakenne. Tähän käytin phpMyAdmin-sovelusta, jolla hallitaan tietokantoja, (kuva 9).



Kuva 9. phpMyAdmin

Valittuani tietokantani "keyo\_test01" lisäsin sinne taulukon nimeltä "Customers". Tauluun lisäsin kaksi kenttää. Toinen ilmoittamaan ajan ja toinen kertomaan, kuka datan on lisännyt. Kenttää, jossa kerrotaan, kuka datan on lisännyt, voi käyttää tunnistamaan kenelle tiedot kuuluvat. Tässä kohdassa on myös helppo asettaa rajoituksia ja määritteitä sille, minkä tyyppistä dataa mihinkin kohtaan voi lisätä. Esimerkiksi "time"-sarakeeseen annoin tyyppiä "TIMESTAMP", jonka ansiosta

se tunnistaa sisältämänsä tiedon tietyntyypillisesti ilmoitetuksi ajankohdaksi. Taulukon luonti on esitetty kuvassa 10.

Taulun nimi: Customers Lisää 1 column(s) Siirry

| Nimi | Tyyppi    | Pituus/Arvot | Oletusarvo | Aakkosjärjestys | Attribuutit | Tyhjä | Indeksi | A_I | Kommentit |
|------|-----------|--------------|------------|-----------------|-------------|-------|---------|-----|-----------|
| time | TIMESTAMP |              | None       |                 |             |       | ---     |     |           |
| user | VARCHAR   | 50           | None       |                 |             |       | ---     |     |           |

Taulun kommentit: Collation: Storage Engine: MyISAM

PARTITION definition:

Partition by: ( Expression or column list )

Partitions:

Esikatsela SQL Tallenna

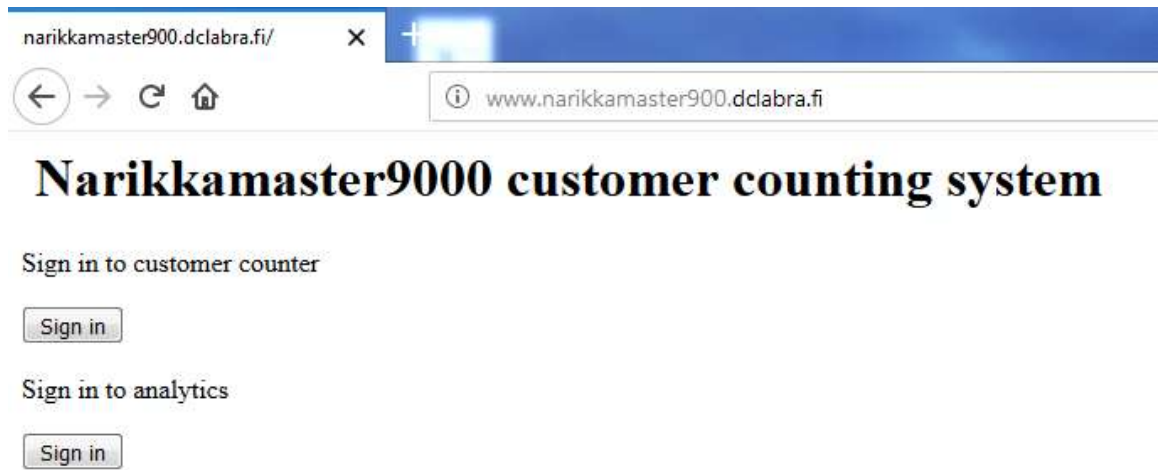
Kuva 10. Taulukon luonti

Taulukon asetuksia pystyy myöhemmin muuttamaan tarpeen vaatiessa, mutta nykyiset asetukset riittävät ympäristön pystyttämiseen. Kun kaikki tarpeellinen tieto on annettu, tallennetaan taulukko tietokantaan.

### 3.3.1.2 Verkkosivun luonti

Nyt kun tietokannassa on taulukko, johon pystyy tallentamaan kerättyä tietoa, pitää siihen tehdä käyttäjäliittymä, jolla lisätä tietoa sinne. Tähän käytin hyvin yksinkertaista sivurakennetta, jossa on mahdollista kirjautua sisään ja lisätä merkintä tietokantaan yhtä nappia painamalla. Sivun rakensin käyttämällä HTML- sekä PHP skriptikieltä.

Etusivulla on kaksi kirjautumispainiketta, joista toisesta pääsee käsiksi asiakasmäärien lisäämissivustolle ja toisesta tarkkailemaan kerättyä dataa, (kuva 11).



Kuva 11. Etusivu

Asiakasmäärän laskentaan kirjautuessa käyttäjä syöttää aiemmassa kohdassa luodun käyttäjänimen ja salasanan tekstikenttiin ja painaa "login"-nappulaa. Kirjautumisikkuna toimii suunnitellusti, mutta oikeaan käyttöön otettaessa lähetetty tieto olisi salattava. Tämänhetkisessä järjestelmässä kunnollista salausta ei ole vielä tehty. Kirjautuminen on esitetty kuvassa 12.



Kuva 12. Asiakaslaskentaan kirjautuminen

Kirjaututtuaan järjestelmään käyttäjälle aukeaa ikkuna, jossa hän voi lisätä tietokantaan merkinnän saapuneesta asiakkaasta yksinkertaisesti painamalla "+1"-painiketta. Kun käyttäjä haluaa lopettaa laskentajärjestelmän käytön, voi hän kirjautua ulos ja palata etusivulle. Sivun esitys on kuvassa 13.



Kuva 13. Asiakaslaskenta

### 3.2.2 Tiedon käyttö

Nyt kun tietokantaan pystyy lisäämään tietoa, tarvitaan järjestelmään osuus, jossa informaatiota pystytään tarkastelemaan käytännöllisesti. Tätä varten tarvitaan jonkinlainen hakutoiminto sekä graafinen kuvaaja, jossa näyttää haettua tietoa.

Tämäkin osuus on toteutettu selainpohjaisesti hyvin yksinkertaisella käyttäjäympäristöllä. Jotta tiedon esittäminen selaimessa olisi käytännöllistä, lisäsin PHP tiedostoihin kuvaajien tekoa varten suunnitellun kirjaston JpGraphin[4].

Etusivulta mennään käyttäjätiedon analysoinnin kirjautumissivulle. Käyttäjämäärien analysointiin kirjaudutaan samalla tavalla kuin kävijämäärälaskuriinkin, (kuva 14).



narikkamaster900.dclabra.fi/loginar X +

← → ↻ 🏠 ⓘ www.narikkamaster900.dclabra.fi/loginanalytics.php


username:

password:

login

Kuva 14. Käyttäjämääräanalysointiin kirjautuminen

Kun käyttäjä on kirjautunut analysointisivustolle, täytyy hänen seuraavaksi valita ajanjakso, jolta haluaa tarkastella kävijämääriä. Hakutoiminnossa valitaan vuoden, kuukauden, päivän ja tunnin mukaan ajanjakso, jolta kävijämäärät halutaan nähdä. Valittuaan halutun aikavälin painaa käyttäjä "Show" painiketta nähdäkseen tulokset. Hakusivu on esitetty kuvassa 15.



narikkamaster900.dclabra.fi/analyti X +

← → ↻ 🏠 ⓘ www.narikkamaster900.dclabra.fi/analytics.html

Choose the timeframe to inspect

From:

Year | Month | Day | Hour

2018 ▼ May ▼ 3 ▼ 20 ▼

To:

Year | Month | Day | Hour

2018 ▼ May ▼ 4 ▼ 03 ▼

Show

Kuva 15. Hakukenttä

Mikäli tuloksia ei ole lainkaan, näkee käyttäjä vain tiedon siitä, että yhtään tulosta ei löytynyt. Muussa tapauksessa käyttäjä pääsee katsomaan tuloksia painamalla uudestaan "Show"-painiketta, (kuva 16).



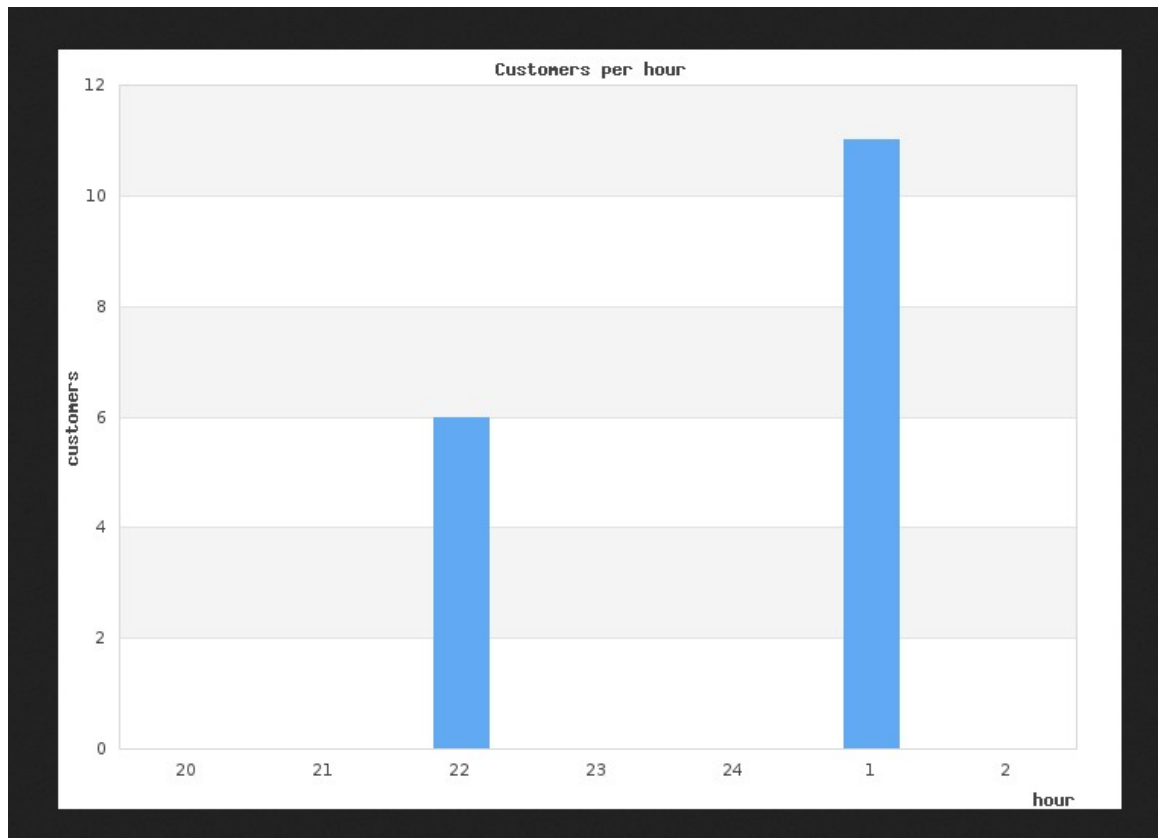
Kuva 16. Ei tuloksia

Tässä vaiheessa ilmenee koko järjestelmän hyöty. Mikäli tiedot vain olisi tallennettu johonkin listaan, olisi niiden selaaminen epäkäytännöllistä jopa hakutoiminnon jälkeen. Esimerkiksi hakukenttäkuvassa esitetyn aikavälin tiedot tietokannassa näyttävät tekstimuodossa kuvan 17 kaltaisilta.

```
2018-05-04 01:45:50 keyo_testuser01@localhost
2018-05-04 01:45:49 keyo_testuser01@localhost
2018-05-04 01:45:43 keyo_testuser01@localhost
2018-05-04 01:08:09 keyo_testuser01@localhost
2018-05-04 01:08:08 keyo_testuser01@localhost
2018-05-04 01:08:07 keyo_testuser01@localhost
2018-05-04 01:07:59 keyo_testuser01@localhost
2018-05-04 01:06:15 keyo_testuser01@localhost
2018-05-04 01:06:15 keyo_testuser01@localhost
2018-05-04 01:06:14 keyo_testuser01@localhost
2018-05-04 01:06:13 keyo_testuser01@localhost
2018-05-03 22:39:56 keyo_testuser01@localhost
2018-05-03 22:39:55 keyo_testuser01@localhost
2018-05-03 22:39:54 keyo_testuser01@localhost
2018-05-03 22:39:53 keyo_testuser01@localhost
2018-05-03 22:39:51 keyo_testuser01@localhost
2018-05-03 22:39:49 keyo_testuser01@localhost
```

Kuva 17. Taulukkoesitys

Hetken tarkastelun jälkeen luettelosta saa selville, että kävijöitä oli 22:00–23:00 kuusi kappaletta ja kello 01:00–02:00 yksitoista kappaletta, mutta graafisen järjestelmän ansiosta kyseinen informaatio selviää nopealla vilkaisulla, (kuva 18).



Kuva 18. Graafinen esitys

### 3.2.3 Tiedon kulku

Tiedon siirtäminen paikasta toiseen muuttui huomattavasti alkuperäisen sovelluspohjaisen toteutuksen vaihduttua selainpohjaiseen. Käytännön toteutuksessa tieto kulkee toimialueen sisäisesti kaikissa tilanteissa.

Toimialueen sisäisen tietoliikenteen ansiosta jotkin tiedon tarkistuskohteet muuttuivat tarpeettomiksi. Esimerkiksi koska käyttäjä ei pääse vaikuttamaan hallitsemattomasti tietokantaan lähetettävään informaatioon, on SQL-injektion torjuminen huomattavasti helpompaa.

Käyttäjän pyytämä tieto siirretään PHP-skriptiä käyttämällä sivulta toiselle ja salasanan ja käyttäjätunnuksen ollessa oikeat tieto lähetetään MySQL-tietokantaan.

## 4 TIETOTURVA

Asiakasmäärätietokannassa säilytettävä tieto on hyödyllistä sekä yritykselle, jolle se kuuluu, että kyseisen toimijan kilpailijoille. Lisäksi laissa on määritetty tietyn-tyyppisten tietojen säilytyksestä tarkkoja säännöksiä.

### 4.1 Lailliset kysymykset

Henkilötietolakia joudutaan soveltamaan järjestelmissä, joissa kerätään tietoa, joka on mahdollista yhdistää henkilöihin. Koska tämän työn järjestelmä kerää vain saapumisajan ja tiedon siitä, että joku on saapunut yrityksen tiloihin, ei ole mitään yksilöitävää tietoa, johon henkilötietolakia tarvitsisi soveltaa [5]. Muut tiedonsäilyt-tämisvelvollisuudetkaan eivät koske tätä järjestelmää, joten varsinaisia laillisia vaatimuksia järjestelmälle ei sen nykyisessä muodossaan ole [6]. Mikäli järjestel-mää mukautettaisiin niin, että sillä voitaisiin kerätä yksilöitäviä henkilötietoja, tar-vitsisi myös lain vaatimat säännökset järjestelmässä täyttää.

### 4.2 Palveluntarjoajan ja -saajan näkökulma

Vaikka laillisia velvoitteita ei järjestelmän turvallisuudelle olisikaan, ei mikään yritys halua mitään tietojaan vahingossa ulkopuolisten käsiin. Näin ollen on järjestelmän tietoturvan vakuutettava asiakas siitä, että heille kuuluva tieto ei päädy kilpailijoille tai muillekaan tahoille. Tämän takia tiedon kulku on oltava salattua ja tietokanta suojattuna.

Jopa hyvin yksinkertaisista tiedoista voi olla paljon hyötyä kilpailijoille. Esimerkiksi tieto siitä, että milloin johonkin yökerhoon saapuu eniten asiakkaita, auttaa ajoitta-maan omaa toimintaansa viedäkseen asiakkaat itselleen. Tämä on kaikkein yksin-kertaisin esimerkki mahdollisesta haitasta, joka aiheutuisi siitä, että joku pääsisi hänelle kuulumattomiin tietoihin käsiksi.



Palveluntarjoajalle järjestelmän tietoturvalla ei sinänsä ole niin suurta merkitystä, sillä järjestelmä on yksinkertaisuutensa ansiosta helppo korjata jonkin mennessä vikaan. Tietenkin kaikkoavat asiakkaat ovat palveluntarjoajan ongelma, mutta se ongelma lähtee asiakkaalle aiheutuvasta ongelmasta.

#### 4.3 Mahdollisia riskejä ja niiden torjuntaa

Järjestelmän turvaamiseksi on tehtävä tietoturvariskien kartoitus, jossa pohditaan ja mitoitetaan mahdollisia uhkia ja niiden vaikutuksia. Parhaimmillaan järjestelmään saadaan vakaita ratkaisuja, jotka turvaavat sen monilta uhilta samaan aikaan. Esimerkiksi tiedon kahdennus moneen eri fyysiseen sijaintiin ehkäisee monia vahinkoihin liittyviä vaaroja.

Ulkoisten pilvipalveluiden käyttö on realistisin ja kannattavin tapa varmistaa tiedon säilyvyys. Tässä pitää kuitenkin ottaa huomioon, että tiedot ovat nyt jonkun muun säilytyksessä. Vaikka laillisia ongelmia tämän järjestelmän keräämän tiedon säilytyksessä ei olekaan, niin on kuitenkin ensisijaisen tärkeää valita luotettava pilvipalveluiden tarjoaja.

Ulkoisen säilytystilan käyttö lisää tiedon siirtymistä paikasta toiseen, joten itse tiedon kulun turvaaminen on tässäkin yhteydessä turvattava. Yrityksen sisäisen verkon liikenteeseen on huomattavasti työläämpi päästä käsiksi kuin internetin välityksellä kulkevaan tietoon. Silti kyseinen malli on tiedon säilyvyyden ja kustannusten kannalta järkevin.

Tahallinen vahingonteko on toinen tärkeä tietoturvaan liittyvä asia, joka on huomioitava. Osa tällaisesta vahingonteosta koettaa hyötyä omistetusta tiedosta, mutta on olemassa myös sattumanvaraista ilman erityistä syytä tehtävää haittaa.

#### 4.4 Tietokannan varmuuskopiointi

Tietokannan varmuuskopiointi on tärkeää monista syistä. Siihen kohdistuvat riskit ovat yksi syy, mutta esimerkiksi tiedon arkistointi ja kopioiminen ovat myös syitä

varmuuskopioinnille. Tietokannan vaarantuminen voi aiheutua monista syistä. Luonnonkatastrofit, inhimilliset virheet ja laitteiston hajoaminen voivat kaikki vaurioittaa tietokantoja. Varmuuskopiointi pitää tehdä eri laitteelle, kuin missä tietokanta sijaitsee. Muutoin laitteiston hajotessa menetetään samalla sekä tietokanta että sen varmuuskopio. [7]

Tietokannan kahdentaminen moneen sijaintiin ei välttämättä ole kustannustehokasta, jos sen tekee käyttäen omia resursseja. Esimerkiksi ylimääräisen tilan vuokraaminen vain tätä tarkoitusta varten ei pienimuotoisessa toiminnassa ole järkevää.

## LÄHTEET

- (1) Behmen Aida. Tested Happy Hour Ideas That Will Boost Your Catering Business. 2017; Available at: <https://possector.com/marketing-promotion/happy-hour-ideas>.
- (2) Liiketoiminnan sähköistyminen. 2015; Available at: [http://tilastokeskus.fi/til/icte/2015/icte\\_2015\\_2015-11-26\\_kat\\_005\\_fi.html](http://tilastokeskus.fi/til/icte/2015/icte_2015_2015-11-26_kat_005_fi.html).
- (3) Acunetix. What is SQL-injection (SQLi) and how to fix it. 2017; Available at: <https://www.acunetix.com/websitesecurity/sql-injection/>.
- (4) jpgraph. Available at: <https://jpgraph.net/>.
- (5) Henkilötietolaki. 1999; Available at: <https://www.finlex.fi/fi/laki/ajantasa/1999/19990523>.
- (6) Arkistolaki. 1994; Available at: <https://www.finlex.fi/fi/laki/ajantasa/1994/19940831>.
- (7) Back Up and Restore of SQL Server Databases. 2018; Available at: <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/backup-and-restore-of-sql-server-databases?view=sql-server-2017>.